A diffusion-based method for removing background stars from astronomical images

A Diffusion-Based Method for Removing Background Stars from Astronomical Images

Maureen Cohen[a,*], Weiping Lu[a]

[a]*School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh Campus, Edinburgh EH14 4AS, United Kingdom*

Abstract

Image processing techniques for background star removal from astronomical images have important applications such as streak detection and star/galaxy differentiation in galaxy morphology classification. Previous star removal methods relied on frame differencing and so required multiple images. In this work we present a new diffusion-based method that can remove stars and noise at the same time, using only a single image. The method measures a parameter called the local asymmetry around each pixel to identify stars and remove them with varying diffusion strength according to the measured parameter. To optimise the balance between signal preservation and star removal, it further incorporates a local stopping function that can adaptively turn the diffusion process on and off by monitoring the image residues. In order to test the method, we constructed a synthetic image that mimics astronomical images of interest and compared the processed images to the ground truth through the measurement of a range of metrics such as mean average error, root mean squared error, peak signal-to-noise ratio and structural similarity index. The results showed clear improvements in all the metrics compared to previous diffusion algorithms. Finally, we applied our method to a dataset of natural images and confirmed its effectiveness at preserving a wide variety of astronomical objects.

*Corresponding author:
Email address: s1144983@ed.ac.uk (Maureen Cohen)
Current address: School of Geosciences, Crew Building, The University of Edinburgh, Alexander Crum Brown Road, Edinburgh EH9 3FF, United Kingdom

Keywords: techniques: image processing, differential equations, astronomy, methods: numerical

1. Introduction

Background stars in images can obscure features of interest or interfere with automatic image processing. Past attempts at removing background stars have sought to reveal nebulae structures obscured by stars [1], avoid contamination of galaxy detections [2], or improve automatic streak detection algorithms [3]. Removing stars can also be a pre-processing step before more sophisticated image manipulation and analysis (e.g. before morphology unveiling [4]). In recent years, automatic image processing has become indispensable for managing the increasing size of data sets in astronomy [5] [6]. Cutting-edge telescopes can now generate terabytes of data a night, and even commercial digital cameras have outputs in the hundreds of gigabytes every night [6].

Automatic galaxy morphology classification seeks to detect galaxies in data returned by digital sky surveys. A combination of human and machine analysis is used to identify and classify galaxies, but the quantity of data has made wholly automated processing and machine learning a priority [7]. However, machine learning methods can misclassify galaxies as stars and vice versa [8], which calls for a pre-processing step in which foreground or background stars are removed. This has prompted a search for star/galaxy differentiation methods through techniques such as mathematical morphology [9], multi-level clustering [10], and decision trees [11].

Streak detection is used to find and track meteors, near-Earth asteroids, satellites, and space debris. Computer vision methods used in streak detection, notably the Hough transform, can have a high rate of false positives if the images are not pre-processed. False positives result from misinterpreting collinear rows of stars, unusually large or bright stars, or distorted stars found at the edges of wide-angle lenses. These difficulties are enhanced when attempting to detect streaks fainter than the starry background. Pre-processing such

images with a star removal algorithm is a promising avenue for avoiding false positives and improving detection rates and accuracy. Common methods for reducing false detections are mean subtraction or frame differencing, but both have drawbacks [12]. As the point-spread function of stars varies from frame to frame, subtraction-based methods introduce noise and artefacts [13]. These methods work best when the images being subtracted are taken close together in time [14] [15]. Differencing of images taken far apart in time is also a commonly used processing step, but requires careful astrometric alignment and matching of PSFs, even when using established and successful methods [16]. Another proposed star removal method identifies known stars from the Sloan Digital Sky Survey, but this method can not be applied to images outside catalog coverage and performance is highly dependent on catalog completeness [17]. All existing methods rely on information outside the image itself, whether from additional images or from a star catalogue.

In this paper, we present a novel star removal method using only the information contained in a single image. This approach can be used even when a reference image is not available or cannot be adequately calibrated. Our method is realised through a diffusion equation-based, iterative algorithm. Diffusion equations have long been used in image processing to smooth away Gaussian noise; here, we formulate a new diffusion coefficient that selectively locates and smooths stars. The algorithm measures a quantity called the local asymmetry in a small region around each pixel to pinpoint stars. To protect meaningful image features from blurring, the algorithm applies a new, adaptive signal-preserving local stop. The local stop monitors the image residues to detect when a feature begins to appear and 'turns off' diffusion at these pixels. We find that such a local stopping approach is highly effective at preserving even wispy, narrow, faint, and irregular shapes while allowing smoothing to continue in other, still noisy regions of the image.

To evaluate this new method, we constructed a synthetic image to allow comparison of a processed image to a ground truth. The mean average error and root mean squared error between the processed image and the ground truth

were used to evaluate the fidelity of the output. The peak signal-to-noise ratio and structural similarity index were calculated to determine the quality of the processed image. We then applied the method to a test data set of images of nebulae with diverse morphologies and degrees of corruption by noise and background stars. Our method was able to completely remove background stars while preserving the image features with a high degree of sharpness and contrast.

2. Method

2.1 Diffusion equations in image processing. Diffusion equations are used in image processing primarily for denoising and deblurring [18]. A diffusion filter for denoising was first implemented in a seminal paper by Perona and Malik in 1990 [19]. They showed that an image convolved with a Gaussian kernel of a given σ is also a solution of the diffusion equation, which can be obtained by iterative numerical modelling. The general diffusion equation in two dimensions is:

$$\frac{\partial u}{\partial t} = c \cdot (\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) \tag{1}$$

Here, c is the diffusion coefficient and determines the speed of diffusion. The diffusion coefficient need not be constant and can be chosen to vary spatially, so that diffusion is enhanced in some regions and suppressed in others. In the Perona-Malik diffusion filter, the coefficient depends on the local image gradient ΔI, interpreting large gradients as image features (edges) and small gradients as noise. It denoises by diffusing only small gradients, while protecting edges. Perona and Malik's insight has been adapted for diverse applications by designing c to achieve specific outcomes [20] [21] [22] [23] [24] [25] [26].

2.2 A novel diffusion coefficient for star removal. In this paper, we develop the Perona-Malik model in two respects in order to adapt it to removing background stars. We first propose a diffusion coefficient as a function of local asymmetry to capture key features we wish to smooth away: stars. Secondly, we introduce a local stopping function to preserve image features which, due to their diverse

and irregular morphology, may suffer blurring after repeated iterative diffusion. Our coefficient takes the form:

$$c(x, y, t) = \frac{1}{1 + A_S \cdot S \cdot \exp(\frac{I_{avg}}{b_{avg}} - 1)} \tag{2}$$

where A_S is the asymmetry parameter, S is the local stopping function, and the exponential term is an enhancement factor that enhances or diminishes the effect of the coefficient in select regions of the image. We introduce the parameter A_S based on the observation that there are always large gradients in all directions around the peak, so that we can use this geometric analysis to quantify the shape of a star. As the diffusion coefficient is arranged to be inversely proportional to A_S, blurring only occurs in regions of high symmetry (low asymmetry). Smoothing is selectively applied to stars, while other image features are left largely untouched. The exponential function $\exp(\frac{I_{avg}}{b_{avg}} - 1)$ modifies the measured local asymmetry based on other local features. In astronomical images, stars can be found both in dark background regions and shining through objects, such as nebulae, which we do not wish to blur. As the local asymmetry identifies stars in both contexts, a diffusion-based algorithm would remove them equally if there were no parameters to distinguish them, inevitably blurring the features of interest in the image. To avoid over-smoothing the regions of interest, we segment an image into signal and background regions by the ratio $\frac{I_{avg}}{b_{avg}}$ in the exponential, where I_{avg} is the average local intensity, while b_{avg} is the average background intensity of the entire image. If the ratio for a given pixel is high, the pixel is in a disproportionately bright region indicating a nebula, galaxy, meteor, or satellite streak. In this scenario, the large ratio strongly enhances the local asymmetry, decreasing the diffusion coefficient and slowing diffusion. A ratio of unity indicates that the reference pixel is in a background region and does not affect A_S. In this way, smoothing is enhanced in low-information background regions compared to signal regions.

Finally, we introduce the novel local stopping function, S. The local stop addresses a weakness common to diffusion-based image processing filters: the loss of semantically meaningful information that is not quantifiable by spatial

metrics like the image gradient, including subtle textures, irregular or diffuse shapes, and soft edges. Diffusion coefficients are usually functions of spatial information at the present iteration; here, we incorporate information from the past to further improve diffusion-based image processing. Specifically, we use S to stop diffusion in regions of intermingled nebula and background stars when features of interest have begun to be smoothed away. To do so, we denote the image at iteration n as I^n and define the relative image residue as:

$$|I^n - I^{n-1}| \tag{3}$$

Therefore, relative residue is what has been removed from the image since the previous iteration of the algorithm. If a pixel in the relative residue is judged to be part of a semantically meaningful or "signal" region, the local stop $S \to \infty$, the diffusion coefficient $\to 0$, and diffusion is stopped at this pixel for the next iteration. If the pixel is judged to be noise, $S = 1$ and the pixel will be diffused as normal in the next iteration. The local stop can thus be regarded as a pixel- and iteration-specific "off switch." In this way, the local stop continuously monitors the past performance of the algorithm and adapts the diffusion effect at each iteration. The exact implementation will be discussed in the following section.

3. Implementation

3.1 Discretising the diffusion equation. The Perona-Malik diffusion method is described as anisotropic because it depends on the image gradient, which varies with direction. As our coefficient depends on a measure of symmetry, however, we use an isotropic discretisation:

$$I_{i,j}^{n+1} = I_{i,j}^n + \gamma \cdot c \cdot (\nabla_N I + \nabla_S I + \nabla_E I + \nabla_W I)_{i,j}^n \tag{4}$$

Here, I is a grayscale image with pixel indices given by i, j, at the number of iterations n. The constant, γ, is defined as the step size and must be $\leq \frac{1}{4}$ for the numerical model to be stable. The diffusion coefficient c is given by Eq. 2 above. The indices N, S, E, W stand for north, south, east, and west, and

indicate the four nearest neighbours of a given pixel i, j. The image gradients ∇I are defined simply as the difference in the grey values of the reference pixel and each of its neighbours:

$$\begin{aligned} \nabla_N I_{i,j} &= I_{i-1,j} - I_{i,j} \\ \nabla_S I_{i,j} &= I_{i+1,j} - I_{i,j} \\ \nabla_E I_{i,j} &= I_{i,j+1} - I_{i,j} \\ \nabla_W I_{i,j} &= I_{i,j-1} - I_{i,j} \end{aligned} \tag{5}$$

3.2 Local asymmetry. Figure 1 shows how we define the local asymmetry, which can be expressed as a function of ∇I. The proximity of the pixels used to calculate ∇I determines the window size or scale of the calculation. For example, for a given central pixel i, j, the gradient to each of its four nearest neighbours in the N, S, E, W directions is calculated. The gradients along the x- and y-axes are subtracted from one another, respectively, and the resulting differences are summed to form the asymmetry parameter for the window size 1. This process can be repeated for the four second-nearest neighbours in the N, S, E, W directions and indeed an arbitrary number of times at different scales and averaged such that:

$$a = \frac{\Sigma_{p=1}^{q}(|\nabla_{E,p} I - \nabla_{W,p} I| + |\nabla_{N,p} I - \nabla_{S,p} I|)}{q} \tag{6}$$

where the index p represents the window size defining the scale. The maximum window size, q, should be chosen to be just larger than the largest star in the image.

The factor a can be normalised to account for the typical asymmetries of a given image. This is done by dividing a by a threshold parameter k equal to, for example, twice the average value of a over the entire image, calculated based on the initial unprocessed image:

$$A_S = \frac{a}{k} \tag{7}$$

If a star is an ideal point source, the intensity gradients in opposite directions (east-west or north-south) should be roughly equal. The difference between

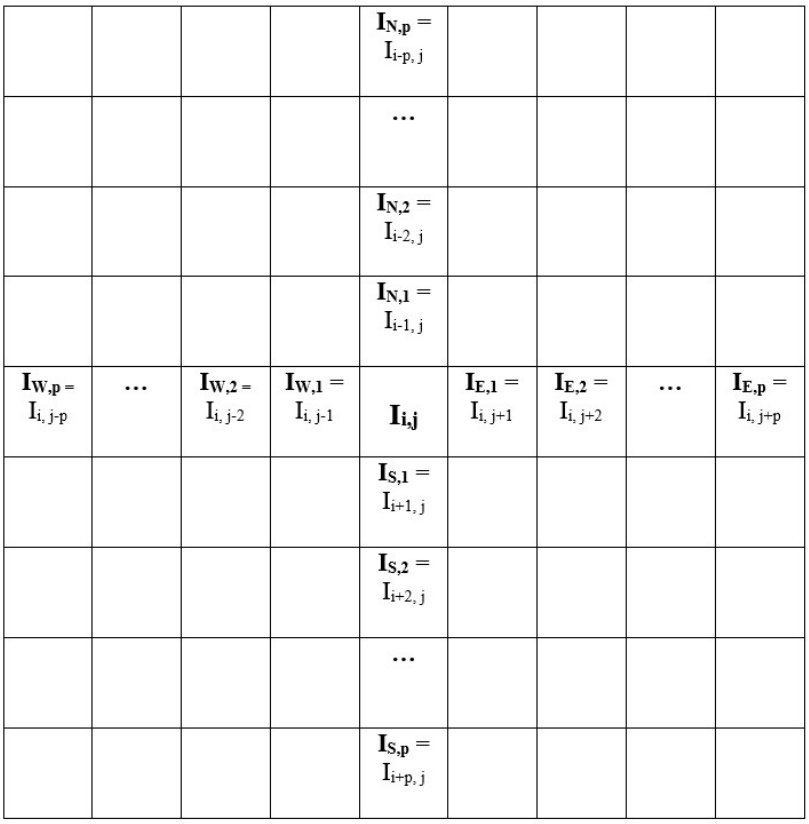


Figure 1: Pixels of the same index p are subtracted from one other and their differences are summed to calculate the local asymmetry. The window size can be chosen to match the typical star sizes of the image or data set.

them will be around zero and the value of A_S very small. In contrast, if the intensity gradients are very unbalanced, as they may be at a feature edge, the local asymmetry will be large. As the diffusion coefficient is inversely proportional to A_S, areas of small asymmetry such as stars will be strongly diffused, while irregular areas with greater asymmetry such as edges will not be affected. We introduced different window sizes to capture the fact that stars may have a peak larger than a single pixel and natural images frequently have stars of multiple sizes.

3.3 Enhancement factor. In the enhancement factor $\exp(\frac{I_{avg}}{b_{avg}} - 1)$, I_{avg} is the average intensity in a window around the reference pixel i, j, while b_{avg} is the average background intensity of the image as a whole and must be calculated empirically for each input image or data set before processing. The window size for calculating the average local intensity should be as small as possible to ensure the enhancement factor is sensitive to small-scale intensity changes in the image. The region chosen to calculate b_{avg} should be an area containing only stars and dark background, without astronomical objects or features of interest.

3.4 Local stop. Like the enhancement factor, the local stop S can be mathematically formulated as an exponential and will take values of either ∞ or 1. For $S \to \infty$, the diffusion coefficient is 0 and diffusion stops. For $S = 1$, the diffusion coefficient does not change.

We implement the local stop using an intermediate binary image, B, which takes values of 0 or 1 and has the same dimensions as the original image. The binary image B is constructed as follows:

1. On each iteration of the algorithm, the relative residue defined by Eq. 3 as $|I^n - I^{n-1}|$ is calculated. To reduce noise, the relative residues of an arbitrary previous number of iterations can be averaged.
2. The averaged relative residue is then thresholded such that pixels with intensity less than or equal to the threshold are set to 0 and pixels with intensity greater than the threshold are set to 1. The threshold is chosen to be the average value of non-background pixels in the relative residue. This ensures that only meaningful image information is used to construct the local stop. The output is the new binary image B, which relates to S through the formula:
$$S = \exp(\frac{1}{B} - 1) \tag{8}$$
Pixels with a B-value of 1 correspond to $S = 1$, while pixels with a B-value of 0 imply that $S \to \infty$.
3. A labelling function is then used to connect groups of pixels with value 1. Once all groups are labelled, another threshold value is applied such that

large groups, which are image features beginning to appear in the residue, are set to 0 and small groups, which are stars, are set to 1. The group size threshold should ideally be chosen to be the size of the largest star in the image. This ensures that objects bigger than the largest star are protected, while those smaller are smoothed.

4. Regions classified as purely background can be set to $B = 1$ for all iterations as there is no need to locally stop diffusion in areas with no features of interest.

We note that the local stop method is broadly adaptable to other applications of the diffusion equation in image processing. Eq. 8 could be multiplied by any given parameter and used as a pixel-specific on/off switch in a variety of situations.

4. Preparation of synthetic data

The performance of denoising algorithms can be tested by adding noise to a clean image, processing the synthetic noisy image, and comparing the output to the original noise-free image, known as the ground truth. To test our method, we modified this approach and constructed a synthetic image by merging a starless image of a nebula with a generic image of the night sky and Gaussian noise. The synthetic image was then processed and compared to the original, starless ground truth.

The model data set for the synthetic image was a collection of photographs from the Australian Astronomical Observatory (formerly the Anglo-Australian Observatory) [27] originally taken by David Malin and archived in digitised form at `https://images.datacentral.org.au/malin/home`. Chao and Tsai [1] previously used this data set to demonstrate a diffusion-based algorithm with a coefficient based on the local grey value variance. We chose this data set to allow a direct comparison of our algorithm to their earlier work. The synthetic image was constructed by the following steps:

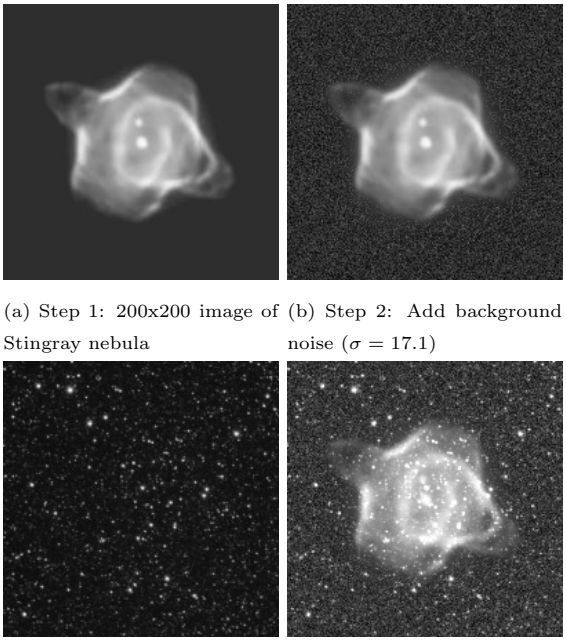
1. The model images were standardised to grayscale and 200x200 pixels in size. Background sections of these images were selected and the average background intensity and standard deviation measured. A new, blank 200x200 pixel image with this background intensity and standard deviation was created to represent background noise.
2. An image of the Stingray nebula taken by the Hubble Space Telescope was superimposed on the synthetic background. We chose this image because it did not contain any other sources. The nebula and background were not added together, but merged such that each pixel of the resulting intermediate image takes the intensity of the brighter of the nebula or background. This ensures that the nebula will have the same brightness in the synthetic image and the ground truth.
3. Finally, a real 200x200 pixel image of a starry night sky was added on top of the intermediate image.

The result is a good visual match to the Australian Astronomical Observatory data set. The image has areas that are solely background and areas where signal and noise are intermingled as stars shine through the nebula. Some areas of the nebula are bright and others are faint and wispy. These diverse features allow the effects of the algorithm on different types of signal to be analysed.

5. Testing on synthetic data

We first describe the parameters and values chosen to process the synthetic image prepared in Fig. 2. They are summarised in Table 1. The local asymmetry, enhancement factor, relative residue, and local stop are also visualised at different iterations for greater clarity. The Perona-Malik diffusion algorithm was implemented and optimised for Python by Alistair Muldal [28]. We used this Python implementation as the basis for our code.

5.1 Visualisation of the diffusion coefficient. We first visualise the key parameters of the diffusion coefficient.



(a) Step 1: 200x200 image of Stingray nebula

(b) Step 2: Add background noise ($\sigma = 17.1$)

(c) Step 3: 200x200 image of stars

(d) Step 4: Add stars to intermediate image

Figure 2: Creation of a synthetic image to test and compare star removal algorithms

Parameter	Meaning	Value
Programming language		Python
Image data type		8-bit (grayscale), $0 - 255$
Image size		$200x200$ pixels
γ	Step size	0.25
a	Local asymmetry	Varies spatially, average of 5 window sizes
k	Normalisation factor	$2x$ average local asymmetry of original, unprocessed image
A_S	$\frac{a}{k}$	Normalised local asymmetry
I_{avg}	Average local intensity	Varies spatially, calculated in 24x24 pixel window
b_{avg}	Average background intensity	Intensity value of 46, for a pixel value range of 0–255
$\lvert I^n - I^{n-1}\rvert$	Relative residue	Varies temporally
S	Local stop	1 or ∞
B	Binary image used to implement local stop	1 or 0
Image segmentation threshold	Value of $\frac{I_{avg}}{b_{avg}}$ used to determine whether an image region is signal or background	2.5
Binary stop threshold	Threshold that determines whether pixels in the binary image are set to 1 or 0	Average intensity of signal areas in the relative residue
Group size	Number of connected pixels to qualify as a feature	Less than 50 is considered a star, 50 or more is considered a feature
n	Number of iterations	200

Table 1: Summary of computational parameters used in experiments

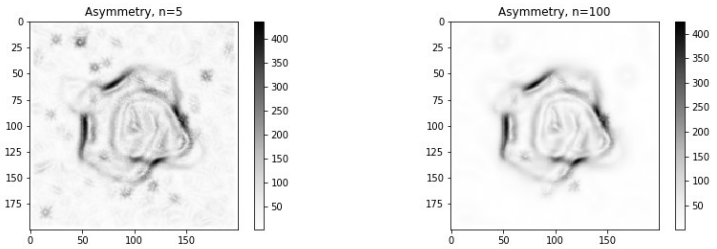


Figure 3: Local asymmetry at $n = 5$ and $n = 100$

5.1.1 Local asymmetry Figure 3 shows the local asymmetry a of the synthetic image at iterations 5 and 100, found by averaging the results of five different window sizes ($q = 5$ in Eq. 6). Dark areas indicate high asymmetry and will be protected from diffusion until feedback from the local stop determines whether they are edges or stars. It can be seen that the largest stars register as having high asymmetry in early iterations – indeed, stars may often have slightly elliptical shapes – but are no longer visible in later iterations as they are smoothed away. Meanwhile, the nebula edges remain as bright and distinct at $n = 100$ as at $n = 5$.

5.1.2 Enhancement factor Figure 4 shows the ratio used in the enhancement factor, $\frac{I_{avg}}{b_{avg}}$. I_{avg} is the average intensity in a $24x24$ window around the reference pixel. Multiple window sizes ranging from $3x3$ to $24x24$ were tested, but we found the result is not sensitive to the window size. The b_{avg} is the average background value of the image, here 46. Darker areas have higher average intensities compared to the background. Areas where the ratio is ≥ 2.5 are classified as signal, while areas where the ratio is < 2.5 are background. This threshold value is chosen empirically for the specific image. Diffusion is slowed in the signal areas.

5.1.3 Local stop Figure 5 shows the relative residue $|I^n - I^{n-1}|$ without implementation of the local stop at two different iterations, $n = 40$ and $n = 100$. While the nebula edges are barely beginning to appear at $n = 40$, they are very

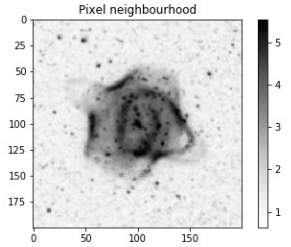


Figure 4: Colormap of ratio $\frac{I_{avg}}{b_{avg}}$

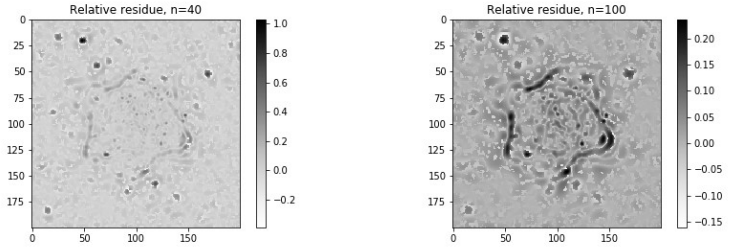


Figure 5: Relative residue at $n = 40$ and $n = 100$

clear at $n = 100$. Without the local stop, these edges would be smoothed away, but the inclusion of S protects the nebula from smoothing.

Figure 6 shows the process flow for constructing the binary image B at iteration $n = 40$. First, the relative residues of the previous four iterations are calculated and averaged to reduce noise. In step two, the relative residue (top left) for the iteration $n = 40$ is thresholded to create the binary image in the top right. The threshold was chosen to be the average intensity value of the signal areas of the relative residue (as represented by the ratio $\frac{I_{avg}}{b_{avg}}$). Pixels brighter than the threshold are set to 1 and pixels fainter than the threshold are set to 0. In step three, the labelling function *skimage.measure.label* from the open-source Python package *scikit-image* labels connected regions of 1's. It identifies large groups (nebula edges) and sets them to 0 to protect them from smoothing. Small groups (stars) remain set to 1 so that diffusion will continue at these pixels. The size threshold for large groups was set to 50 pixels, the

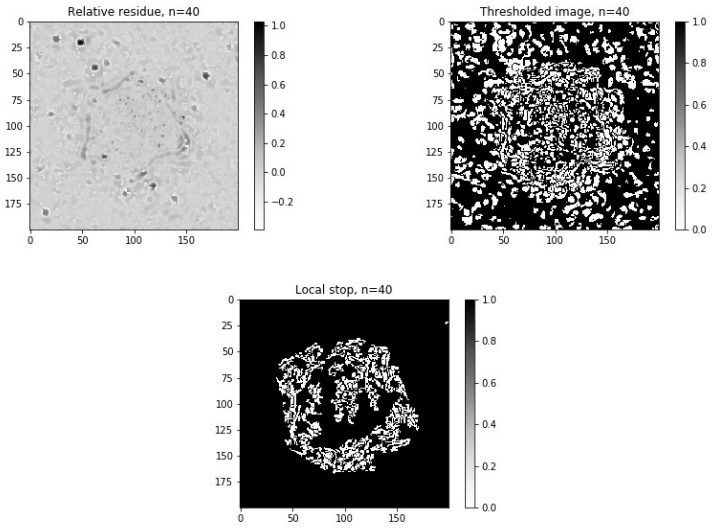
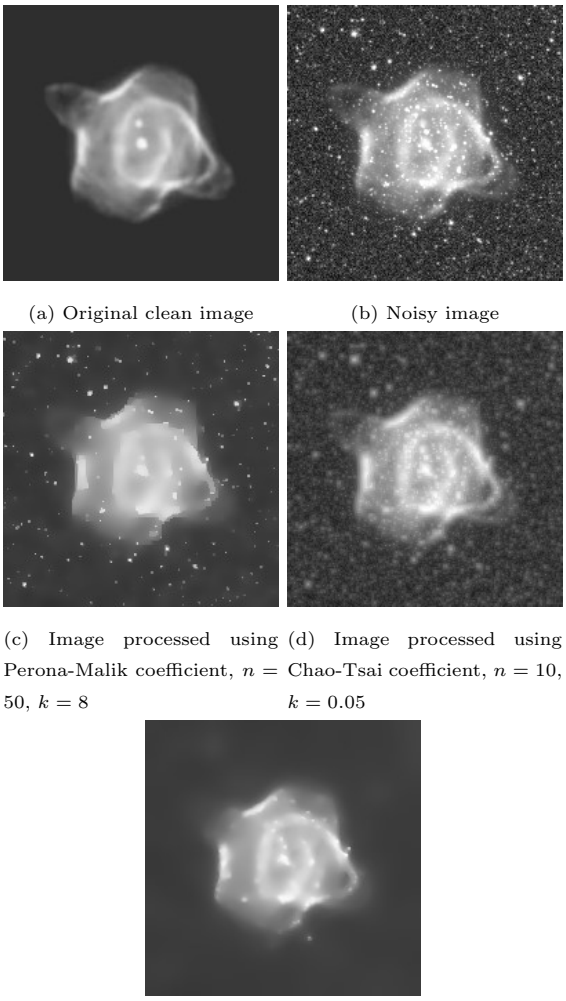


Figure 6: Process flow for constructing the binary image B: relative residue (top left), thresholded image (top right), binary image (bottom). The thresholded image has been reversed ($0 \longleftrightarrow 1$) for ease of visual comparison with the final B.

size of the largest star. Finally, areas classified as background are uniformly set to 1 so that diffusion can proceed in low-information background regions. The binary image is recalculated for each iteration according to the steps shown, effectively turning diffusion "on" or "off" for each pixel depending on whether the algorithm judges the pixel to be signal or star.

5.2 Results of testing on synthetic data. We processed the synthetic image using the Perona-Malik method, the Chao-Tsai method, and our new method and performed quantitative measurements of the results, shown in Table 2: mean absolute error, root mean squared error, peak signal-to-noise ratio of processed image, and structural similarity index when compared with the ground truth, where a score of 1 is a perfect match. Figure 7 gives a visual comparison of the processed images.

We note that the sharp reduction in the image error and improvement in the structural similarity index in our method compared to the Perona-Malik and Chao-Tsai coefficients is attributable to both the complete removal of stars in



(a) Original clean image

(b) Noisy image

(c) Image processed using Perona-Malik coefficient, $n = 50$, $k = 8$

(d) Image processed using Chao-Tsai coefficient, $n = 10$, $k = 0.05$

(e) Image processed using the new method, $n = 200$

Figure 7: Visual results of processing with Perona-Malik, Chao-Tsai, and new method. Here, n is the number of iterations of each algorithm and k is the normalisation constant for the Perona-Malik and Chao-Tsai methods. The values of n and k for the Perona-Malik and Chao-Tsai methods were taken from Chao and Tsai 2006 to allow a rigorous comparison with their work and represent the best visual results from these methods. The value of n for our new method was chosen by trial and error to give the optimal visual result.

250 the background regions and the preservation of contrast in the nebula region. Not only have the stars been removed, sharpness has been maintained and the

Coefficient	Mean absolute error	RMS error	Peak SNR (dB)	Structural Similarity Index
Perona-Malik	233.4	19.72	22.231	0.729
Chao-Tsai	222.5	19.52	22.322	0.476
New method	15.4	15.96	24.068	0.909

Table 2: Image quality measurements for three diffusion-based algorithms

image has not become washed out.

6. Tests on natural images

A selection of images from the David Malin data set has been processed and shown in Figure 8 below to demonstrate the new algorithm's performance on a diverse group of natural images. The left-hand row shows the original image, the centre row the image processed with the Chao-Tsai coefficient, and the right-hand row the image processed with our method.

Our method shows clear improvements in all images studied. In some cases, such as Henize 70, delicate features in the central structure that were barely visible in the original image have been strongly enhanced. In NGC 2014, the processed image gives a better idea of which parts of the image are nebula and which are background stars. Also notable are the lack of blurring of edges in the bright region inside NGC 3576 and the preservation of highly textured and wispy details of the Veil nebula by our method compared to the others.

7. Conclusion and further work

We proposed a new diffusion-based method with an asymmetry-detecting coefficient and adaptive switching for removing background stars. Our tests show that the new method is effective at removing both stars and background

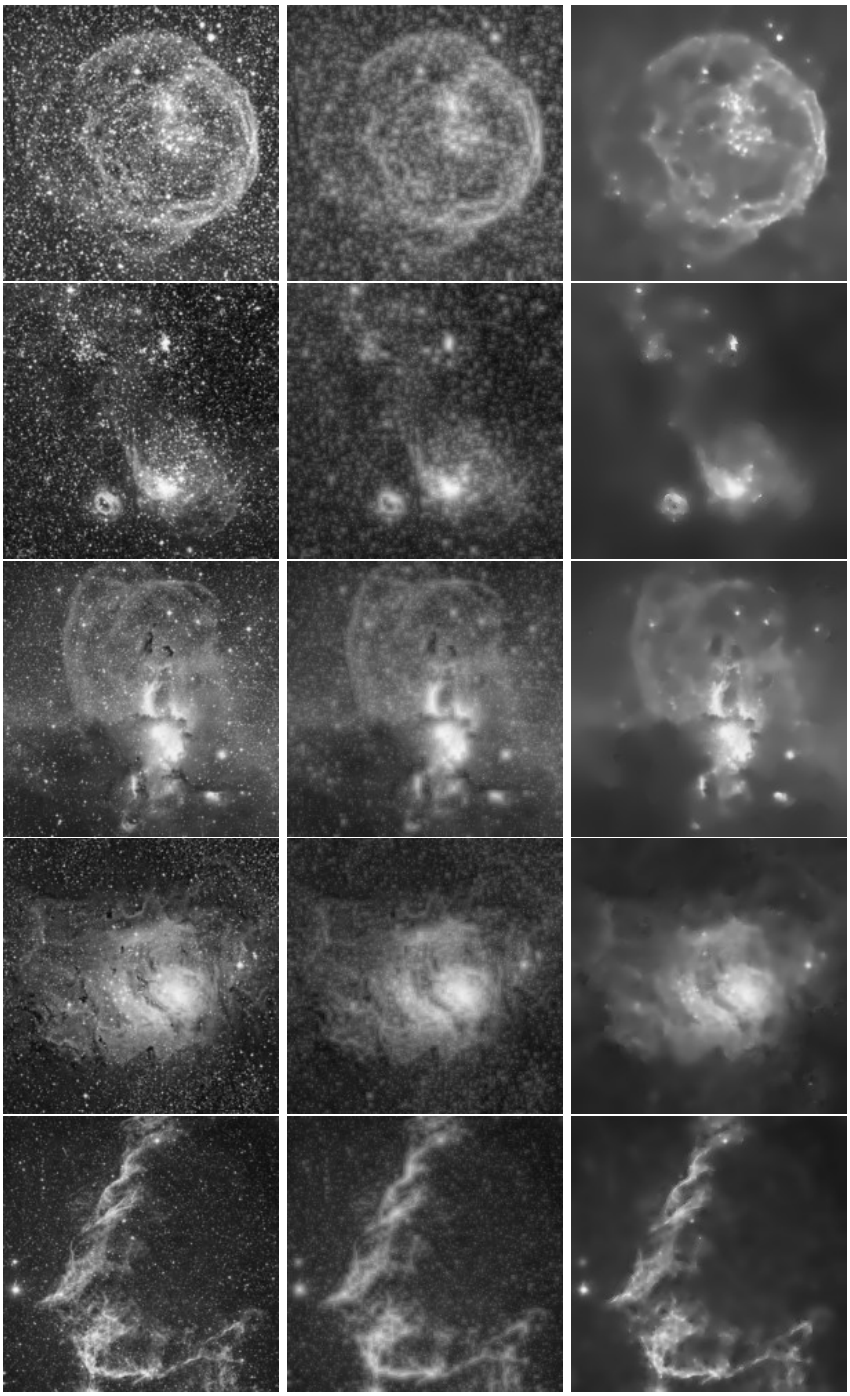


Figure 8: Original image (left column), Chao-Tsai coefficient (centre column) with $n = 10$ for all images, our method (right column) with n as stated in parentheses. Rows from top to bottom: Henize 70 ($n = 200$), NGC 2014 ($n = 100$), NGC 3576 ($n = 300$), NGC 6523 (Lagoon nebula) ($n = 50$), NGC 6995 (Veil nebula) ($n = 30$)

noise while preserving thin, irregular, and low-intensity structures in nebulae. These results indicate that diffusion-based methods are a promising candidate for background star removal applications in astronomy, avoiding the problems of frame differencing methods and allowing star removal in situations when a sequence of images is not available. We note finally that, as in other applications of the diffusion equation in image processing, the algorithm lacks a natural final stopping time,a much-discussed problem in the literature [29] [30] [31]. The run time for the processed images shown above varies by an order of magnitude, from n=30 to n=300, as images with denser and brighter stars required more iterations. Although several ways of determining the final stopping time were tested, none returned a satisfactory outcome, and the issue requires further research and experimentation.

Appendix A. Supplementary material

The source code for this algorithm is available at `https://github.com/maureenjcohen/diffusion-for-stars`.

References

[1] S. Chao, D. Tsai, Astronomical image restoration using an improved anisotropic diffusion, Pattern Recognition Letters 27 (2006) 335–344. doi:https://doi.org/10.1016/j.patrec.2005.08.021.

[2] Z. Frei, Semi-automatic removal of foreground stars from images of galaxies, Publications of the Astronomical Society of the Pacific 108 (1996) 624. doi:10.1086/133775.
URL http://dx.doi.org/10.1086/133775

[3] G. Nir, B. Zackay, E. Ofek, Optimal and efficient streak detection in astronomical images, The Astronomical Journal 156 (2018) 229–244. doi:https://doi.org/10.3847/1538-3881/aaddff.

[4] M. Pesenson, W. Roby, B. McCollum, Multiscale astronomical image processing based on nonlinear partial differential equations, The Astrophysical Journal 683 (2008) 566–576. doi:https://doi.org/10.1086/589276.

[5] N. Ball, R. Brunner, Data mining and machine learning in astronomy, International Journal of Modern Physics D 19 (7) (2010) 1049–1106. doi:https://doi.org/10.1142/S0218271810017160.

[6] J. Kremer, K. Stensbo-Smidt, F. Gieseke, K. Pedersen, C. Igel, Big universe, big data: Machine learning and image analysis for astronomy, IEEE Intelligent Systems 32 (2017) 16–22. doi:https://doi.org/10.1109/MIS.2017.40.

[7] E. Kuminski, J. George, J. Wallin, L. Shamir, Combining human and machine learning for morphological analysis of galaxy images, Publications of the Astronomical Society of the Pacific 126 (944) (2014) 959–967.
URL http://www.jstor.org/stable/10.1086/678977

[8] Clarke, A. O., Scaife, A. M. M., Greenhalgh, R., Griguta, V., Identifying galaxies, quasars, and stars with machine learning: A new catalogue of classifications for 111 million SDSS sources without spectra, Astronomy and Astrophysics 639 (2020) A84. doi:http://10.1051/0004-6361/201936770.
URL https://doi.org/10.1051/0004-6361/201936770

[9] J. A. Moore, K. A. Pimbblet, M. J. Drinkwater, Mathematical morphology: Star/galaxy differentiation and galaxy morphology classification, Publications of the Astronomical Society of Australia 23 (04) (2006) 135–146. doi:http://doi.org/10.1071/as06010.
URL http://dx.doi.org/10.1071/AS06010

[10] H. Du, Y. Wang, C. Ren, J. Zhong, X. Gao, Multi-level clustering algorithm for star/galaxy separation, in: 2016 12th International Conference on Computational Intelligence and Security (CIS), 2016. doi:https://doi.org/10.1109/ICIP.2010.5651365.

[11] E. C. Vasconcellos, R. R. de Carvalho, R. R. Gal, F. L. LaBarbera, H. V. Capelato, H. F. C. Velho, M. Trevisan, R. S. R. Ruiz, Decision tree classifiers for star/galaxy separation, The Astronomical Journal 141 (6) (2011) 189. doi:10.1088/0004-6256/141/6/189.
URL https://doi.org/10.1088%2F0004-6256%2F141%2F6%2F189

[12] S. Molau, P. Gural, A review of video meteor detection and analysis software, WGN, Journal of the International Meteor Organization 33 (1) (2016) 15–20.

[13] B. Zackay, E. Ofek, A. Gal-Yam, Proper image subtraction—Optimal transient detection, photometry, and hypothesis testing, The Astrophysical Journal 830 (27) (2016). doi:https://doi.org/10.3847/0004-637X/830/1/27.

[14] P. Gural, Algorithms and software for meteor detection, Earth, Moon, and Planets 102 (2007) 269–275. doi:https://doi.org/10.1007/s11038-007-9161-7.

[15] T. Suk, S. Šimberová, Automated meteor detection by all-sky digital camera systems, Earth, Moon, and Planets 120 (2017) 189–215. doi:https://doi.org/10.1007/s11038-017-9511-z.

[16] C. Alard, R.H. Lupton, A method for optimal image subtraction, The Astrophysical Journal 503 (1) (1998) 325–331. doi:https://doi.org/10.1086/305984.

[17] D. Bektešević, D. Vinković, Linear feature detection algorithm for astronomical surveys - I. Algorithm description, MNRAS 471 (3) (2017) 2626–2641. doi:https://doi.org/10.1093/mnras/stx1565.

[18] J. Koenderink, The structure of images, Biological Cybernetics 50 (5) (1984) 363–370. doi:https://doi.org/10.1007/BF00336961.

[19] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (7) (1990) 629–639. doi:https://doi.org/10.1109/34.56205.

[20] J. Weickert, Applications of nonlinear diffusion in image processing and computer vision, Acta Math. Univ. Comenianae LXX (1) (2001) 33–50.

[21] A. Buades, B. Coll, J.-M. Morel, Image enhancement by non-local reverse heat equation, Tech. Rep. 22, CMLA (2006).

[22] Y. Yu, S. Acton, Speckle reducing anisotropic diffusion, IEEE Transactions on Image Processing 11 (11) (2002) 1260–1270. doi:https://doi.org/10.1109/TIP.2002.804276.

[23] G. Gilboa, N. Sochen, Y. Zeevi, Forward-and-backward diffusion processes for adaptive image enhancement and denoising, IEEE Transactions on Image Processing 11 (7) (2002) 689–703. doi:https://doi.org/10.1109/TIP.2002.800883.

[24] P. Guidotti, Y. Kim, J. Lambers, Image restoration with a new class of forward-backward-forward diffusion equations of Perona-Malik type with applications to satellite image enhancement, SIAM J. Imaging Sci. 6 (3) (2013) 1416–1444. doi:https://doi.org/10.1137/120882895.

[25] L. Wang, S. Luo, Z. Wang, Image deblur with regularized backward heat diffusion, in: IEEE 17th International Conference on Image Processing, 2010. doi:https://doi.org/10.1109/ICIP.2010.5651365.

[26] X. Zhang, X. Feng, Texture-preserving Perona-Malik model, in: 4th International Congress on Image and Signal Processing, 2011, pp. 812–815. doi:https://doi.org/10.1109/CISP.2011.6100263.

[27] D. Malin, David Malin Archive of the Anglo-Australian Observatory, https://images.datacentral.org.au/malin/home (2020).

[28] A. Muldal, Implementation of Perona-Malik diffusion in python, https://pastebin.com/sBsPX4Y7 (2012).

[29] A. Ilyevsky, E. Turkel, Stopping criteria for anistropic PDEs in image processing, Journal of Scientific Computing 45 (2010) 333–347. doi:https://doi.org/10.1007/s10915-010-9361-6.

[30] P. Mrázek, M. Navara, Selection of optimal stopping time for nonlinear diffusion filtering, International Journal of Computer Vision 52 (2003) 189–203. doi:https://doi.org/10.1023/A:1022908225256.

[31] C. Tsiotsios, M. Petrou, On the choice of parameters for anisotropic diffusion in image processing, Pattern Recognition 46 (5) (2013) 1369–1381. doi:https://doi.org/10.1016/j.patcog.2012.11.012.